IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicants	:	Kaushik Saha et al.
Application No.	:	10/727,138
Filed	:	December 3, 2003
For	:	LINEAR SCALABLE FFT/IFFT COMPUTATION IN A MULTI-PROCESSOR SYSTEM

Examiner	:	Chat C. Do
Art Unit	:	2193
Docket No.	:	852463.406
Date	:	June 21, 2010

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

DECLARATION OF DR. KAUSHIK SAHA AND MR. SRIJIB NARAYAN MAITI UNDER 37 C.F.R. § 1.132

Commissioner for Patents:

We, Dr. Kaushik Saha and Mr. Srijib Narayan Maiti, hereby declare that:

1.   Dr. Kaushik Saha is a co-inventor of the above-referenced patent application ("the present application") and is a researcher and electrical engineer employed by STMicroelectronics Asia Pacific Pvt. Ltd., the assignee of the present application. Dr. Saha has over 20 years of experience in the field of designing signal processors and embedded signal processing systems, and is considered an expert in the field of signal processing. A copy of Dr. Saha's Curriculum Vitae is attached hereto as Appendix 1.

2.   Mr. Srijib Narayan Maiti is a co-inventor of the present application and is a researcher and electrical engineer employed by STMicroelectronics Asia Pacific Pvt. Ltd., the assignee of the present application. Mr. Maiti has over 12 years of experience in the field of

designing signal processors and embedded signal processing systems, and is considered an expert in the field of signal processing. A copy of Mr. Maiti's resume is attached hereto as Appendix 2.

3.   We have reviewed the present application, the Office Action mailed February 1, 2010, including the objections to the drawings and the rejections under 35 U.S.C. § 112, first paragraph, for lack of enablement, and under 35 U.S.C. § 103(a) for obviousness, and the references cited therein, and the objection to the Amendment of November 12, 2009, as allegedly introducing new matter. We submit this Declaration to the Examiner in support of the patentability of the presently claimed invention, which is generally directed to methods of computing Fast Fourier Transforms (FFT) or Inverse Fast Fourier Transforms (IFFT) of digital signals in multiprocessor systems.

Objections to the Drawings

4.   We understand the Examiner objected to the drawings for failing to show the "structure of computing/performing N-point FFT/IFFT of the signal using first and second stages wherein the second stage employs single, un-nested computational loop." We understand that none of the claims uses the exact language which the Examiner contends is not shown in the drawings. Accordingly, we discuss the independent claims with reference to supporting figures below.

5.   We understand that independent claim 1, as amended, recites, "[a] method of processing a digital signal by computing a Fast Fourier Transform (FFT) or Inverse Fast Fourier transform (IFFT) of the digital signal, the method comprising using a multiprocessor computing system having a plurality of processors P configured to perform the steps of: computing an N-point FFT/IFFT of the signal using first and second sets of butterfly computational stages, each stage in the second set of stages employing a plurality of butterfly operations, wherein each of the butterfly operations in each stage in the second set of stages has a single, un-nested computation loop; and distributing the plurality of butterfly operations in each stage of the second set of stages such that each processor computes an equal number of complete butterfly operations thereby eliminating data interdependency among the parallel processors." To the extent the Examiner is referring to the multiprocessor system, an embodiment of such a

2

system is show in Figure 4, where a multiprocessing system comprises clusters 12 (*i.e.*, processors, see Specification at 7, line 24 to 8, line 25). Figure 5 contains more detail on an embodiment of a cluster of Figure 4. To the extent the Examiner is referring to the recited method steps, Figures 2 and 3 illustrate implementations of embodiments of the method. Figure 2 shows a two-processor implementation and Figure 3 shows a four processor implementation, where different line styles represent computation in each of the processors. See Page 6, line 25 to 7, line 22 of the specification as filed.

6.   We understand that independent claim 3 recites, "[a] multiprocessor system to process a digital signal by computing a Fast Fourier Transform (FFT) or Inverse Fast Fourier transform (IFFT) of the signal using a decimation in time or decimation in frequency approach, comprising: the multiprocessor system having a plurality of processors P and configured to implement: means for computing a first plurality of $\log_2 P$ stages of an N-point FFT/IFFT of the signal; means for computing a second plurality of stages of the N-point FFT/IFFT of the signal using in each stage of the second plurality of stages a plurality of butterfly operations, wherein each butterfly operation employs a single butterfly computation loop without employing nested loops; and means for distributing the butterfly operations in each stage of the second plurality of stages such that each processor computes an equal number of complete butterfly operations thereby eliminating data interdependency in the stages of the second plurality of stages." To the extent the Examiner is referring to the multiprocessor system, an embodiment of such a system is show in Figure 4, where a multiprocessing system comprises clusters 12 (*i.e.*, processors, see Specification at 7, line 24 to 8, line 25). Figure 5 contains more detail on an embodiment of a cluster of Figure 4. To the extent the Examiner is referring to the recited functions, Figures 2 and 3 illustrate embodiments of the implementation of the functions. Figure 2 shows a two-processor implementation and Figure 3 shows a four processor implementation, where different line styles represent computation in each of the processors. See Page 6, line 25 to 7, line 22.

7.   We understand that independent claim 5, as amended, recites, "[a] non-transitory computer-readable storage medium whose contents cause a system having a plurality of processors to perform a linear scalable method of transforming a signal by computing with the plurality of processors a Fast Fourier Transform (FFT) or an Inverse Fast Fourier Transform

(IFFT) of the signal, the method comprising: computing a first plurality of stages of an N-point FFT/IFFT; and computing a second plurality of stages of the N-point FFT without employing nested loops and by distributing the butterfly operations in each stage of the second plurality of stages such that each processor computes an equal number of complete butterfly operations thereby eliminating data interdependency in the stage." To the extent the Examiner is referring to a system having a plurality of processors, an embodiment of such a system is show in Figure 4, where a multiprocessing system comprises clusters 12 (*i.e.*, processors, see Specification at 7, line 24 to 8, line 25). Figure 5 contains more detail on an embodiment of a cluster of Figure 4. To the extent the Examiner is referring to the recited method, Figures 2 and 3 illustrate the implementation of embodiments of the method. Figure 2 shows a two-processor implementation and Figure 3 shows a four processor implementation, where different line styles represent computation in each of the processors. See Page 6, line 25 to 7, line 22.

8.   We understand that independent claim 16, as amended, recites, "[a] non-transitory computer-readable storage medium whose contents cause a system having a plurality of processors to perform a linear scalable method of transforming a signal, the method comprising: computing an N-point FFT/IFFT using a first plurality of butterfly computational stages and a second plurality of butterfly computational stages, each stage in the second plurality of stages employing a plurality of butterfly operations having a single, un-nested computation loop; and distributing the plurality of butterfly operations in each stage of the second plurality of stages such that each processor computes an equal number of complete butterfly operations thereby eliminating data interdependency in the stage." To the extent the Examiner is referring to a system having a plurality of processors, an embodiment of such a system is show in Figure 4, where a multiprocessing system comprises clusters 12 (*i.e.*, processors, see Specification at 7, line 24 to 8, line 25). Figure 5 contains more detail on an embodiment of a cluster of Figure 4. To the extent the Examiner is referring to the recited method, Figures 2 and 3 illustrate the implementation of embodiments of the method. Figure 2 shows a two-processor implementation and Figure 3 shows a four processor implementation, where different line styles represent computation in each of the processors. See Page 6, line 25 to 7, line 22.

9.   We understand that independent claim 27, as amended, recites, "[a] method of transforming a digital signal, the method comprising: using a multiprocessor

4

computing system having a plurality P of processors configured to: compute a first number of butterfly stages of an N-point Fast Fourier Transform (FFT) or Inverse Fast Fourier transform (IFFT); and compute remaining butterfly stages of the N-point FFT/IFFT with a single iterative loop wherein each processor computes an equal number of butterfly operations and there is no data dependency between butterflies in a stage of an iteration of the loop." To the extent the Examiner is referring to a multiprocessor computing system having a plurality of processors, an embodiment of such a system is show in Figure 4, where a multiprocessing system comprises clusters 12 (*i.e.*, processors, see Specification at 7, line 24 to 8, line 25). Figure 5 contains more detail on an embodiment of a cluster of Figure 4. To the extent the Examiner is referring to the recited method, Figures 2 and 3 illustrate the implementation of embodiments of the method. Figure 2 shows a two-processor implementation and Figure 3 shows a four processor implementation, where different line styles represent computation in each of the processors. See Page 6, line 25 to 7, line 22.

   10. We understand that independent claim 31 recites, "[a] system, comprising: an instruction fetch cache; and a plurality of processors P coupled to the instruction fetch catch and configured to: compute a first number of butterfly stages of an N-point Fast Fourier Transform (FFT) or Inverse Fast Fourier Transform (IFFT) of a digital signal; and compute remaining butterfly stages of the N-point FFT/IFFT with a single iterative loop wherein there is no data dependency between butterflies in a stage of an iteration of the loop." To the extent the Examiner is referring to a system comprising an instruction fetch cache and a plurality of processors, an embodiment of such a system is show in Figure 4, where a multiprocessing system comprises clusters 12 (*i.e.*, processors, see Specification at 7, line 24 to 8, line 25). Figure 5 contains more detail on an embodiment of a cluster of Figure 4. To the extent the Examiner is referring to the configuration of the plurality of processors, Figures 2-5 illustrate the implementation of example embodiments. Figure 2 shows a two-processor implementation, Figure 3 shows a four processor implementation, where different line styles represent computation in each of the processors. Figure 4 shows an N processor implementation. See Page 6, line 25 to 7, line 22.

5

The Claims Are Enabled by the Specification

11.   We understand the Examiner rejected claims 1-7, 11-13, 15-18, 27-29 and 31-33 under 35 USC Section 112, first paragraph, as not enabled. We understand the Examiner cited claim 1 as an example and contends that "first and second sets of butterfly computational stages ... wherein the second stage employs a single un-nested computational loop," was never fully addressed in the summary or the detail in such a way that one of skill in the art would be able to make or use the invention. We disagree.

12.   We are familiar with the level of skill in the art of digital signal processing. After reviewing the specification, one of skill in the art would understand what was meant by an un-nested loop in the context of the claims. As an initial matter, one of skill in the art would understand the difference between nested and un-nested loops in general. Moreover, Figures 2 and 3 show multiple stages of an N-point FFT/IFFT implemented on multiprocessor systems, and the description thereof (see pages 6-8) as well as original claim 2 describe how to implement at least one embodiment of distributing the remaining stages of the butterfly operations among the processors in an un-nested loop (for example, "by assigning to each processor of the multi-processor system respective addresses of memory locations corresponding to inputs and outputs required for each specific butterfly operation assigned to the processor"). See original claim 2. This is sufficient to enable one of skill in the art to practice the claimed invention without undue experimentation, which we understand is all that is required to satisfy the enablement requirement. We believe the disclosure is sufficient to enable the remaining independent claims for similar reasons. For convenience, Figures 2 and 3 of the present application are reproduced below, along with portions of the description thereof on pages 6-7.
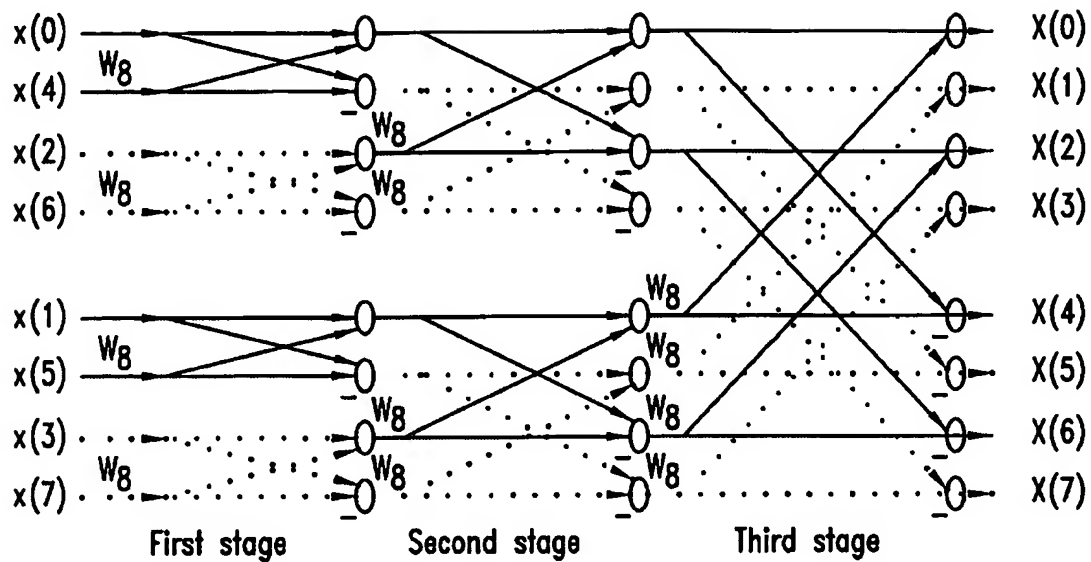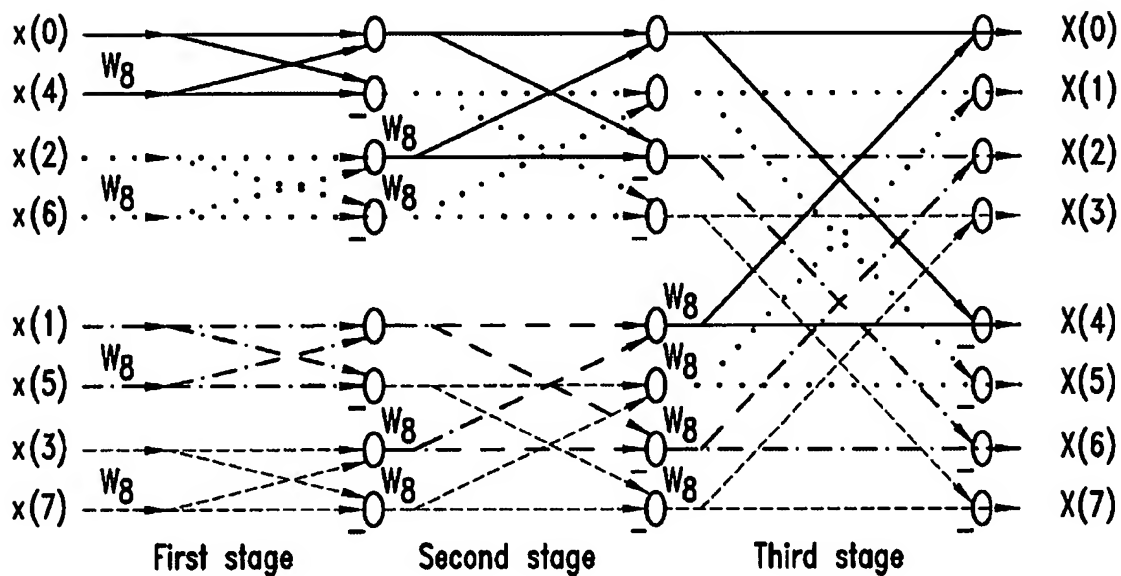
Figure 2 of Present Application

Figure 3 of Present Application

Figure 2 shows the implementation for an 8-point FFT in a 2-processor architecture using the present invention. Dotted lines are computed in one processor, and dashed lines in the other. The computational blocks are represented by 'O'. The left side of each computational block is its input (the time domain samples) while the right side is its output (transformed samples). The present invention uses a mixed radix approach with decimation in time. The first two stages of the radix-2 FFT/IFFT are computed as a single radix-4 stage. As these stages contain only load/stores and add/subtract operations there is no need

for multiplication. This leads to reduced time for FFT/IFFT computation as compared to that with full radix-2 implementation. The next stage has been implemented as a radix-2. The three main nested loops of conventional implementations have been fused into a single loop which iterates $(N/2*(\log_2 N-2))/$(number of processor) times. Each processor is used to compute one butterfly in one loop iteration. Since there is no data dependency between different butterflies in this algorithm, the computational load can be linearly divided among the different processors, leading to the linear scalability.

The mechanism for assigning the butterflies in this manner consists of assigning the memory location to a processor such that each processor computes a complete butterfly. To achieve this a binary digit is inserted at the appropriate bit location in the address of the memory location for input/output data for the computation of the butterfly, depending on the stage of the FFT transformation.

Figure 3 shows a 4-processor implementation for the 8-point FFT using this invention. Different line styles represent computation in each of the 4 processors.

Present Application, Pages 6-7.

Abel, Alone or in Combination with Jaber, Does Not Render the Claims Obvious

13.   We understand the Examiner rejected claims 1-7, 11-13, 15-18, 27-29 and 31-33 under 35 USC Section 103(a) as obvious over U.S. Patent No. 5,991,787 issued to Abel et al., in view of U.S. Patent No. 6,792,441 issued to Jaber. For the reasons set forth below, we do not agree that the combination of Abel and Jaber would render the claimed embodiments obvious to one of skill in the art.

14.   The Examiner appears to rely on Jaber as allegedly teaching the distributing of the butterfly operations. The operation of Jaber for a 16-point FFT distributed among 4 processors is described below with reference to the some of the figures of Jaber, and is contrasted with the operation of embodiments of the present application. This discussion is illustrative purposes only, and can be generalized to larger FFTs (size N), as well as to different numbers of processors.
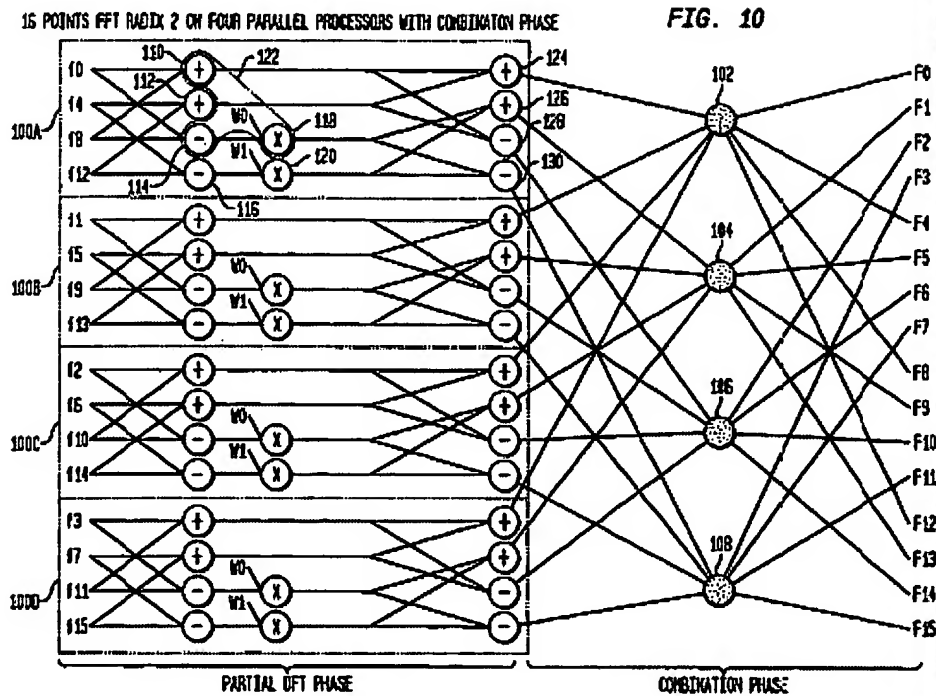
Fig 10 of US6792441 (Jaber et al)

15. With reference to Figure 10 of Jaber, Jaber teaches that the set of input data points (N=16) is divided among 4 processors (P=4) such that each processor works on 4 data points (i.e. executes 2 butterfly computations) in each stage of an FFT. All 4 processors execute this task in parallel without the need to send or receive data to or from the other parallel processor, for the first two stages of FFT. This is followed by a combination phase which needs the outputs (4x4=16) of all these processors and produces 16 outputs finally. The combination phase is comprised of the last 2 (log$_2$P) stages of the FFT.

16. As a result butterfly computations involving input data points are assigned to the processors of Jaber as follows:

{f0, f4, f8, f12} to first processor

{f1, f5, f9, f13} to second processor

{f2, f6, f10, f14} to third processor   G1

{f3, f7, f11, f15} to fourth processor

17. Until stage 2 of the FFT computation, all 4 processors execute their butterfly operations in parallel and independently of each other. Thereafter, stages 3 and 4 of the FFT computation are assigned to a combination phase. For the case of a 16-point FFT, the computations require 8 (N/2) twiddle factors/coefficients named, {W0, ..., W7}. The first two stages require the coefficients W0 and W1 only. The combination phase requires the entire set of coefficients {W0,...,W7}. This necessitates that the coefficients W0, W1 be accessible to all the 4 processors at the same time instant. This is evident from Fig 8 of Jaber, reproduced below, which shows a coefficient memory (804) being accessed by all the processors 807A,...807B.
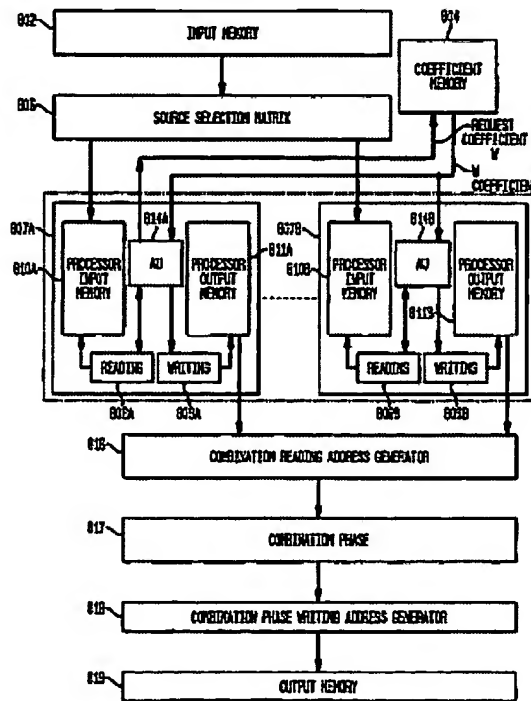

Figure 8 of Jaber

18.   If we consider as a second example the case of a 32 point FFT (N=32) on 4 parallel processors (P=4), input data points are assigned to the processors as follows:

{f0, f4, f8, f12, f16, f20, f24, f28} to first processor

{f1, f5, f9, f13, f17, f21, f25, f29} to second processor

{f2, f6, f10, f14, f18, f22, f26, f30} to third processor   G2

{f3, f7, f11, f15, f19, f23, f27, f31} to fourth processor

19.   There are total 5 stages in this case. Until stage 3 of the FFT computation, all 4 processors execute their butterfly operations in parallel and independently of each other. Thereafter, stages 4 and 5 of the FFT computation are assigned to a combination phase. In this case too, the combination phase is comprised of the last $\log_2 P = 2$ stages of the 32 point FFT.

20.   In this case there are 16 twiddle coefficients {W0,...,W15}, out of which {W0,W1,W2,W3} are required to be accessible to all the 4 processors. The combination phase requires the entire set of coefficients {W0,...,W15}.

21.   In the general case of an N point FFT being executed on P parallel processors, there would be a total of $\log_2 N$ stages, out of which the last $\log_2 P$ stages comprise the combination phase and the remaining ($\log_2 N - \log_2 P$) stages are computed in P processors in parallel. In the general case, there would be N/2 twiddle coefficients in all {W0,...,$W_{N/2-1}$}, out of which the identical N/(2P) number of twiddle coefficients need to be accessible to all the P parallel processors all the while. The combination phase requires access to all the N/2 twiddle coefficients.
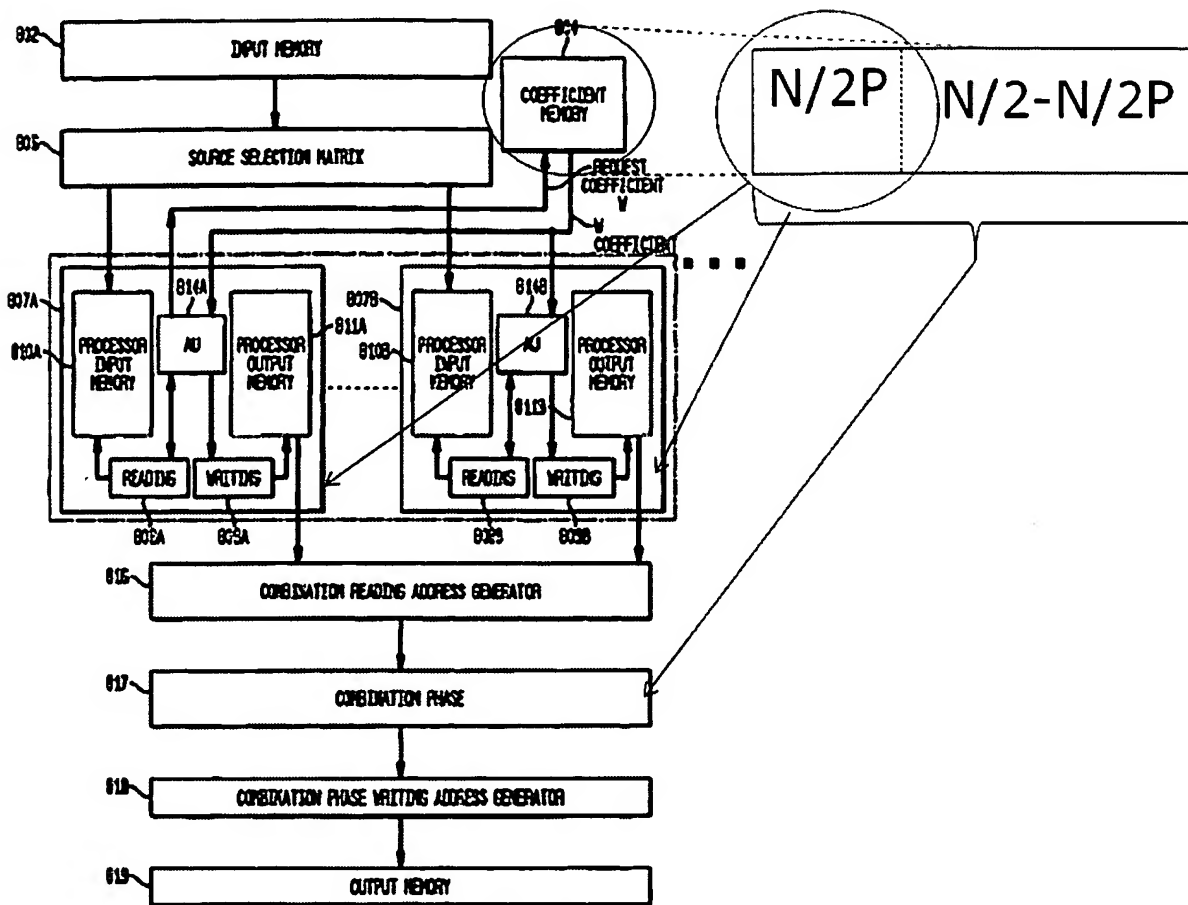
11

Figure 8 of Jaber Modified to Show Distribution of Twiddle Coefficients in Jaber

22.   Figure 8 of Jaber appears again above, modified to show the distribution of the twiddle coefficients from the coefficient memory as per the Jaber. Out of N/2 coefficients, N/2P coefficients are needed by all of the processors all of the time, and the entire set is needed by the combinational phase.
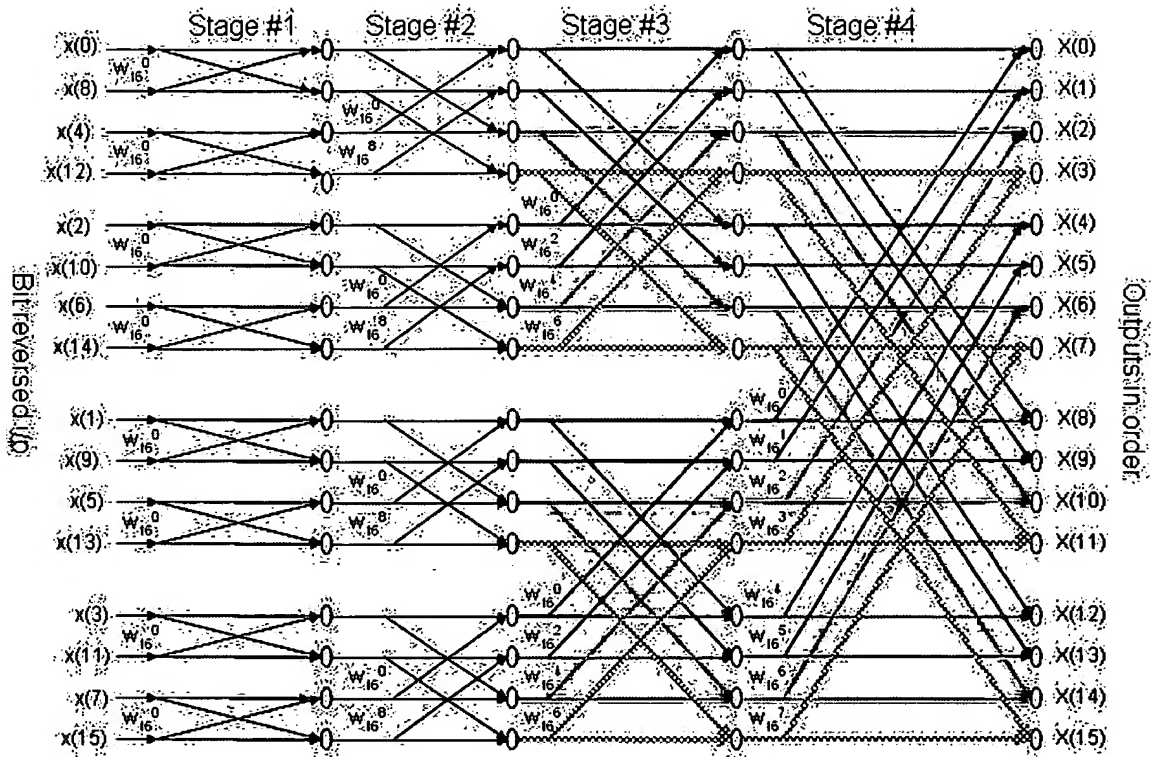
Butterfly Distribution of Present Disclosure for 4-Processor Configuration: N = 16; P = 4

23.   The butterfly distribution of the disclosure of the present application is illustrated above for N = 16 and P = 4. The set of input data points (N=16) is divided among 4 processors (P=4) such that each processor works on 4 data points (i.e. executes 2 butterfly computations) in each stage of the FFT. (A 16 point FFT is shown to facilitate a comparison with Jaber; Figure 3 of the present application shows an 8 point FFT on a 4 processor system). All 4 processors execute this task in parallel without the need to send or receive data to or from the other parallel processors, for the last two stages of the FFT. During the first $\log_2 P = 2$, there is data dependency between the parallel processors, i.e. there is need of data to/from each other in the first two stages. Therefore, parallel processing of butterfly computations starts after the first 2 stages. As shown in the above figure (stage 3, stage 4), the butterflies colored in red are computed in the first processor, the butterflies colored in blue are computed in the second processor, the butterflies colored in green are computed in the third processor, the butterflies colored in black are computed in the fourth processor. It can be seen from stage 4 that all the red colored butterflies take input data points only from red colored butterflies of the previous stage

(i.e. stage 3), and so forth for the other processors. As a result butterfly computations involving input data points are assigned to the processors as follows:

$\{x(0),x(8),x(1),x(9)\}$ to first processor

$\{x(4), x(12), x(5), x(13)\}$ to second processor

$\{x(2), x(10), x(3),x (11)\}$ to third processor   G3

$\{x(6) ,x(14), x(7), x(15)\}$ to fourth processor

     24.   After stage 2 of the FFT computation, all 4 processor execute their butterfly operations in parallel and independently of each other until the final output. Prior to that, stages 1 and 2 of the FFT computation have data dependency among the parallel processors. For the case of a 16-point FFT, the computations require 8 (=N/2) twiddle factors/coefficients $\{W^0,...,W^7\}$. As shown in the butterfly distribution illustrated on the previous page, the requirement of twiddle coefficients by the 4 processors is as follows:

$\{w^0, w^4\}$ to first processor

$\{w^1, w^5\}$ to second processor

$\{w^2, w^6\}$ to third processor   G4

$\{w^3, w^7\}$ to fourth processor

     25.   It may be noted that owing to the innovative scheme of distribution of butterfly computations among processors of the present application, the sets of twiddle coefficients required by different processors are disjoint and the number of coefficients in each set does not exceed 2.

26.   If we consider as a second example the case of a 32 point FFT (N=32) on 4 parallel processors (P=4), input data points are assigned to the processors as follows:

$\{x(0),x(16),x(1),x(17), x(2),x(18),x(3),x(19)\}$ to first processor

$\{x(8),x(24),x(9),x(25), x(10),x(26),x(11),x(27)\}$ to second processor

$\{x(4),x(20),x(5),x(21), x(6),x(22),x(7),x(23)\}$ to third processor   G5

$\{x(12),x(28),x(13),x(29), x(14),x(30),x(15),x(31)\}$ to fourth processor

27.   After stage 2 of the FFT computation, all 4 processor execute their butterfly operations in parallel and independently of each other until the final output. Prior to that, stages 1 and 2 of the FFT computation have data dependency among the parallel processors. For the case of a 32-point FFT, the computations require 16 (=N/2) twiddle factors/coefficients named $\{W^0,....,W^{15}\}$. The requirement of twiddle coefficients by the 4 processors is as follows:

$\{w^0, w^4, w^8, w^{12}\}$ to first processor

$\{w^1, w^5, w^9, w^{13}\}$ to second processor

$\{w^2, w^6, w^{10}, w^{14}\}$ to third processor   G6

$\{w^3, w^7, w^{11}, w^{15}\}$ to fourth processor

28.   It may be noted again that owing to the innovative scheme of distribution butterfly computations among processors of the present application, the sets of twiddle coefficients required by different processors are disjoint and the number of coefficients in each set does not exceed 4.

29.   In the general case of an N point FFT being executed on P parallel processors, there would be a total of $\log_2 N$ stages, out of which the first $\log_2 P$ stages have data dependency between the parallel processors and the remaining ($\log_2 N- \log_2 P$) stages are computed in P processors in parallel without need of data to/from each other. In the general case, there would be N/2 twiddle coefficients in all $\{W^0,....,W^{N/2-1}\}$, out of which disjoint sets N/(2P) of twiddle coefficients are needed by each of the P parallel processors except when P=2.

15

In this case also (P=2), one processor only needs all the twiddle coefficients, whereas the other one needs only a subset having $N/(2P)$ twiddle coefficients.

30.   The fundamental difference lies in the method of butterfly distribution among the parallel processor as shown in the assignments of butterflies among the parallel processors, G1, G3 for 16-point FFT/IFFT and G2, G5 for 32-point FFT/IFFT. As a result of this, the stages where there are dependencies among the parallel processors are in the first $\log_2 P$ stages as per the present disclosure, whereas, the dependencies among the parallel processors are in the last $\log_2 P$ stages as per Jaber.

31.   Another manifestation of the disclosure of the present application can be seen in the uses of twiddle coefficients in the different parallel processors. As per the present proposal, it may be noted that although all $N/2$ coefficients are used for the entire FFT computation, no parallel processor uses more than an $N/2P$ subset of these, having no common coefficients with any other processors except when P=2. Whereas as per Jaber, an identical $N/(2P)$ twiddle coefficients need to be accessible to all the P parallel processors all the while. The combination phase of Jaber requires access to all the $N/2$ twiddle coefficients. Figure 8 of Jaber is modified again on the following page to show how the twiddle coefficients would be applied if Jaber were modified in accordance with the disclosure of the present application. As can be seen, distinct $N/2P$ subsets of the $N/2$ coefficients in the coefficient memory are used by different parallel processors. At any stage no processor has need of any coefficients which are required by any other processor. Please note that the present disclose is not limited in any way to application to the particular architecture of Jaber. It can be applicable to any system architecture in general, including shared and distributed memory systems. The use of shared coefficient memory taught in Jaber is rendered unnecessary in the present proposal due to the innovative method of distribution of butterflies among the parallel processors.
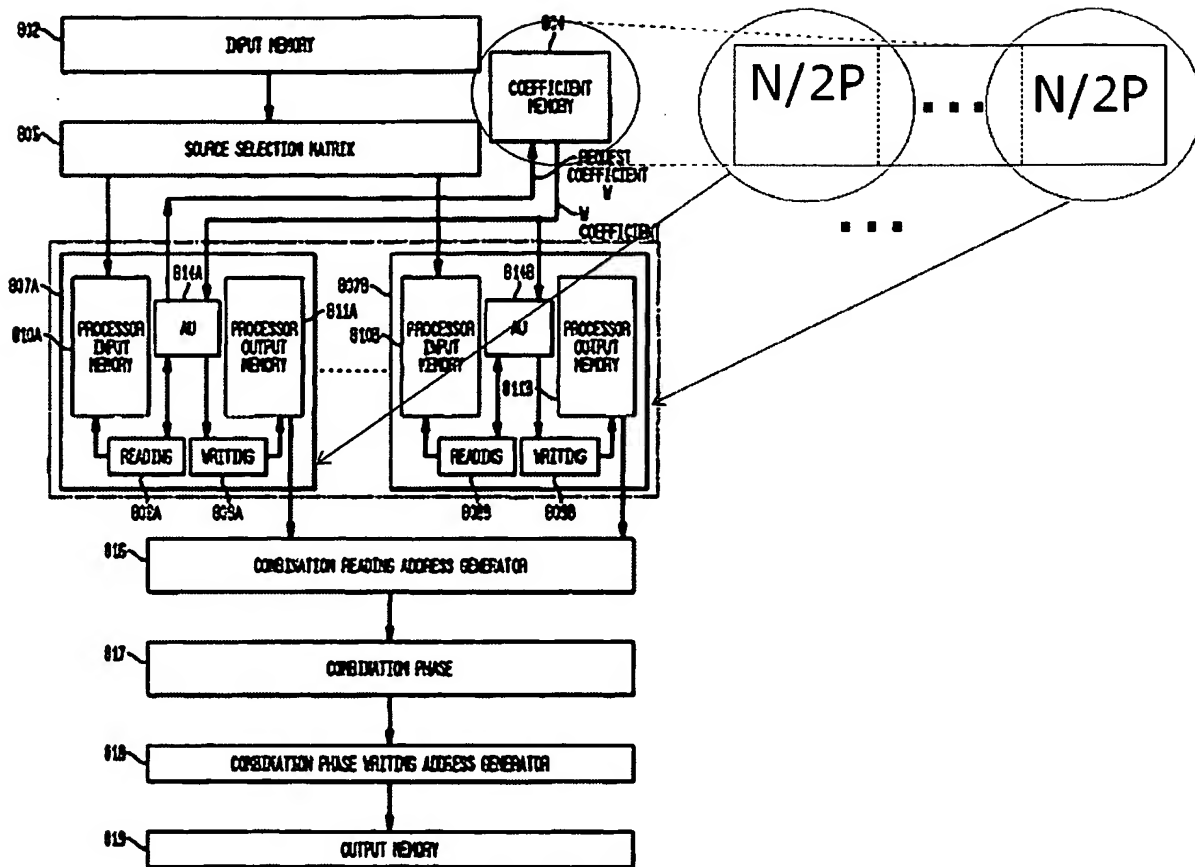
Figure 8 of Jaber Modified to Show Distribution of Twiddle Coefficients
According to Present Disclosure

32.   Turning to the language of the claims, we understand claim 1 as amended recites, "each stage in the second set of stages has a single, un-nested computation loop; and distributing the plurality of butterfly operations in each stage of the second set of stages such that each processor computes an equal number of complete butterfly operations thereby eliminating data interdependency among the parallel processors." Abel, alone or in combination with Jaber, does not teach or suggest, or otherwise render obvious to one of skill in the art, "distributing the plurality of butterfly operations in each stage of the second set of stages such that each processor computes an equal number of complete butterfly operations thereby eliminating data interdependency among the parallel processors," as recited. We understand that claims 2, 7 and 11 depend from claim 1 and are allowable at least by virtue of their dependencies.

17

33. We understand that the Examiner contends that because Jaber stores the coefficients in memory and retrieves the coefficients from memory there is no data dependency. This argument is incorrect because in Jaber, the coefficients are shared between the processors, as discussed above. Specifically, out of N/2 coefficients, N/2P coefficients are needed by all of the processors of Jaber all of the time, and the entire set is needed by the combinational phase. Thus, the stages of Jaber have data interdependencies among the processors.

34. We understand that independent claim 3 recites, "means for computing a second plurality of stages of the N-point FFT/IFFT of the signal using in each stage of the second plurality of stages a plurality of butterfly operations, wherein each butterfly operation employs a single butterfly computation loop without employing nested loops; and means for distributing the butterfly operations in each stage of the second plurality of stages such that each processor computes an equal number of complete butterfly operations thereby eliminating data interdependency in the stages of the second plurality of stages." Abel, alone or in combination with Jaber, does not teach or suggest, or otherwise render obvious to one of skill in the art the recited, "means for distributing the butterfly operations in each stage of the second plurality of stages such that each processor computes an equal number of complete butterfly operations thereby eliminating data interdependency in the stages of the second plurality of stages." As discussed above with respect to claim 1, the stages of Jaber have data interdependencies among the processors. We understand that claims 4, 12, 13 and 15 depend from claim 3 and are allowable at least by virtue of their dependencies.

35. We understand that independent claim 5, as amended, recites, "computing a first plurality of stages of an N-point FFT/IFFT; and computing a second plurality of stages of the N-point FFT without employing nested loops and by distributing the butterfly operations in each stage of the second plurality of stages such that each processor computes an equal number of complete butterfly operations thereby eliminating data interdependency in the stage." Abel, alone or in combination with Jaber, does not teach or suggest, or otherwise render obvious to one of skill in the art, "computing a second plurality of stages of the N-point FFT without employing nested loops and by distributing the butterfly operations in each stage of the second plurality of stages such that each processor computes an equal number of complete butterfly operations thereby eliminating data interdependency in the stage," as recited. As discussed above, the

18

stages of Jaber have data interdependencies among the processors. We understand that claim 6 depends from claim 5 and is allowable at least by virtue of its dependencies.

36.   We understand that independent claim 16, as amended, recites, "computing an N-point FFT/IFFT using a first plurality of butterfly computational stages and a second plurality of butterfly computational stages, each stage in the second plurality of stages employing a plurality of butterfly operations having a single, un-nested computation loop; and distributing the plurality of butterfly operations in each stage of the second plurality of stages such that each processor computes an equal number of complete butterfly operations thereby eliminating data interdependency in the stage." Abel, alone or in combination with Jaber, does not teach, suggest or otherwise render obvious to one of skill in the art, "distributing the plurality of butterfly operations in each stage of the second plurality of stages such that each processor computes an equal number of complete butterfly operations thereby eliminating data interdependency in the stage," as recited. As discussed above, the stages of Jaber have data interdependencies among the processors. We understand that claims 17 and 18 depend from claim 16 and are allowable at least by virtue of their dependencies.

37.   We understand that independent claim 27, as amended, recites, "compute a first number of butterfly stages of an N-point Fast Fourier Transform (FFT) or Inverse Fast Fourier transform (IFFT); and compute remaining butterfly stages of the N-point FFT/IFFT with a single iterative loop wherein each processor computes an equal number of butterfly operations and there is no data dependency between butterflies in a stage of an iteration of the loop." Abel, alone or in combination with Jaber, does not teach, suggest or otherwise render obvious to one of skill in the art, "compute remaining butterfly stages of the N-point FFT/IFFT with a single iterative loop wherein each processor computes an equal number of butterfly operations and there is no data dependency between butterflies in a stage of an iteration of the loop," as recited. as discussed above with respect to claim 1, the stages of Jaber have data interdependencies among the processors. We understand that claims 28 and 29 are allowable at least by virtue of their dependencies.

38.   We understand that independent claim 31, as amended, recites, "compute a first number of butterfly stages of an N-point Fast Fourier Transform (FFT) or Inverse Fast Fourier Transform (IFFT) of a digital signal; and compute remaining butterfly stages of the N-

point FFT/IFFT with a single iterative loop wherein there is no data dependency between butterflies in a stage of an iteration of the loop." Abel, alone or in combination with Jaber, does not teach, suggest or otherwise render obvious to one of skill in the art, "compute remaining butterfly stages of the N-point FFT/IFFT with a single iterative loop wherein there is no data dependency between butterflies in a stage of an iteration of the loop," as recited. As discussed above, the stages of Jaber have data interdependencies among the processors. We understand that claims 32 and 33 are allowable at least by virtue of their dependencies.

No New Matter Has Been Introduced

39.   We understand the Examiner also objected to the amendment filed with the RCE of November 11, 2009 as introducing new matter. Specifically, the Examiner contends each of the independent claims introduces new matter based on the limitations "first and second sets of butterfly computational stages ... wherein the second stage employs single un-nested computational loop."

40.   We understand that the current independent claims do not contain the precise language objected to by the Examiner. Nevertheless, we have attempted to address the Examiner's concerns below.

41.   We understand that independent claim 1, as originally filed, recited:

1.   A linear scalable method for computing a Fast Fourier Transform (FFT) or Inverse Fast Fourier transform (IFFT) in a multiprocessing system using a decimation in time approach, comprising the steps of:

computing first and second stages of $\log_2 N$ stages of an N-point FFT/IFFT as a single radix-4 butterfly operation while implementing the remaining ($\log_2 N$-2) stages using radix-2 butterfly operations, wherein each radix-2 butterfly operation employs a single radix-2 butterfly computation loop without employing nested loops; and

distributing the butterfly operations in each stage such that each processor computes an equal number of complete butterfly operations thereby eliminating data interdependency in the stage.

42.   To the extent the Examiner objects to the introduction of the word "sets" in claim 1 during prosecution, one of skill in the art would have recognized that claim 1 as originally filed was addressed to processing two sets of stages (the first set comprising the first

and second stages of butterfly computational stages and the second set comprising the remaining stages), with the second set (the remaining stages) employing "a single ... computational loop without employing nested loops." A "set" has a well-known meaning to one of skill in the art and one of skill in the art would have recognized that original claim 1 referred to sets of computational stages. Moreover, Figures 2 and 3 illustrates sets of computational stages and the description thereof on pages 6-7 is would have been recognized by one of skill in the art as directed to handling the first two stages in one manner and the remaining stages in another.

43.   To the extent the Examiner is referring to "a single un-nested computational loop" this is almost identical to the original language "a single ... computation loop without employing nested loops." Further, as discussed above, one of skill in the art would have recognized Figures 2 and 3 and the description thereon on pages 6 and 7 as disclosing "computing an N-point FFT/IFFT of the signal using first and second sets of butterfly computational stages, each stage in the second set of stages employing a plurality of butterfly operations, wherein each of the butterfly operations in each stage in the second set of stages has a single, un-nested computation loop."

44.   We understand that independent claim 3, as originally filed, recited:

3.   A linear scalable system for computing a Fast Fourier Transform (FFT) or Inverse Fast Fourier transform (IFFT) in a multiprocessing system using a decimation in time approach, comprising:

means for computing first and second stages of $\log_2 N$ stages of an N-point FFT/IFFT as a single radix-4 butterfly operation while implementing the remaining $(\log_2 N-2)$ stages using radix-2 butterfly operations, wherein each radix-2 butterfly operation employs a single radix-2 butterfly computation loop without employing nested loops; and

means for distributing the butterfly operations in each stage such that each processor computes an equal number of complete butterfly operations thereby eliminating data interdependency in the stage.

45.   Claim 3 as currently submitted does not use the word "sets."

46.   If it is "plurality of stages" to which the Examiner objects in claim 3, "plurality" has a definite meaning in the claims of a patent, specifically, more than one, and one

of skill in the art would have recognized both "first and second stages" as a first plurality of stages, and "remaining stages" as a second plurality of stages.

47.   If it is "employs a single ... butterfly computational loop without employing nested loops," to which the Examiner objects, this language appears in claim 3 as originally filed.

48.   We understand that independent claim 5, as originally filed, recited:

5.   A computer program product comprising computer readable program code stored on a computer readable storage medium embodied therein for computing a Fast Fourier Transform (FFT) or Inverse Fast Fourier transform (IFFT) in a multiprocessing system using a decimation in time approach, comprising:

computer readable program code means configured for computing computing first and second stages of $\log_2 N$ stages of an N-point FFT/IFFT as a single radix-4 butterfly operation while implementing the remaining ($\log_2 N$-2) stages using radix-2 butterfly operations, wherein each radix-2 butterfly operation employs a single radix-2 butterfly computation loop without employing nested loops; and

computer readable program code means configured for distributing the butterfly operations in each stage such that each processor computes an equal number of complete butterfly operations thereby eliminating data interdependency in the stage.

49.   Claim 5 as amended does not use the word "sets."

50.   If it is "plurality of stages" to which the Examiner objects, "plurality" has a definite meaning in the claims of a patent, specifically, more than one, and one of skill in the art would have recognized both "first and second stages" as a first plurality of stages, and "remaining stages" as a second plurality of stages.

51.   If it is "without employing nested loops," to which the Examiner objects, this language appears in claim 5 as originally filed.

52.   With regard to independent claim 16, to the extent the Examiner objects to the word "plurality," "plurality" has a definite meaning in the claims of a patent, specifically, more than one, and one of skill in the art would have recognized both "first and second stages" (see original claim 5) as a first plurality of stages, and "remaining stages" (see original claim 5) as a second plurality of stages. To the extent the Examiner objects to "having a single, un-nested

computational loop," please see original claims 1, 3 and 5, and the discussion of claims 1, 3 and 5 above.

53. With regard to independent claims 27 and 31, to the extent the Examiner objects to "a first number of butterfly stages" one of skill in the art would have recognized "first and second stages" (see original claim 5) as a first number of stages, and "remaining stages" (see original claim 5) as "remaining stages." To the extent the Examiner objects to "a single iterative loop," the concept of a single iterative loop is discussed on page 7, lines 8-13.

54. We hereby declare that all statements made herein of our own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.
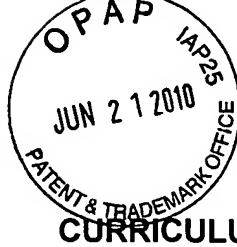
16 / June / 2010
Date

Kaushik Saha
Dr. Kaushik Saha

_14 / 06 / 2010_
Date

Srijib Narayan Maiti
Mr. Srijib Narayan Maiti

APPENDIX 1

CURRICULUM-VITAE OF KAUSHIK SAHA

NAME: Dr. KAUSHIK SAHA

ADDRESS FOR
CORRESPONDENCE:

A-2 Staff Flats,
I. P. College,
Shamnath Marg,
Delhi-110054
India
ph:+91-11-2392-9931
cell: +91-98110-64398
email: kaushik.saha@st.com

DATE OF BIRTH: 14th April, 1965

EDUCATIONAL QUALIFICATIONS:

DEGREE	INSTITUTION	DISCIPLINE	YEAR
B.Tech.	Indian Institute Of Technology, Delhi	Electrical Engineering	1983-87
M.Tech.	Indian Institute Of Technology, Delhi	Computer Technology (Dept. Of Elect. Engg.)	1988-89
Ph.D.	Indian Institute Of Technology, Delhi	Electrical Engineering	1990-96

EXPERIENCE:

Research:

1)   M.Tech. dissertation work was done in speech recognition using the neural network computing paradigm. This required extensive application of digital signal processing.

2)   Subsequently, I have studied issues concerning realization of neural networks using I.C. technology for my PhD thesis. I have looked at design issues in using analog circuits and have designed a useful 4-quadrant analog multiplier which is compact and can be used for diverse applications like signal processing and neural networks.

Industrial:

At STMicroelectronics (Since 1996 till date, descending chronologically)

Currently, I am the Design Unit Manager and Technical Mentor for the Advanced Systems Technology group of ST-Noida, which is the systems

research wing of the company. My group is engaged in advanced processor architecture and application development for various multimedia platforms of STMicroelectronics and ST-Ericsson. I have been with AST since 2001.

1)   I lead teams of Advanced Systems Technologies (AST) group which are working on the design of digital audio and video encoders/decoders on advanced processor architectures (VLIW and SIMD).

2)   The AST group led by me also works in the area of digital multimedia algorithm development on parallel (MIMD) architectures. We are currently engaged in porting and developing the Google Android stack on the ARM SMP based MontBlanc platform from ST-Ericsson, over the Linux-SMP operating system. The group is also engaged in development of Power Management strategies for the multi-core MontBlanc platform over the Linux-SMP operating system. These projects are executed for our clients ST-Ericsson.

3)   I lead the development of advanced video IPs and algorithms. My group is currently engaged in the development of the Context Adaptive Binary Arithmetic Coder (CABAC) hardware IP for the Fairbanks platform from ST-Ericsson, using High Level Synthesis. We are also developing a decoder architecture for the next generation video compression standard called Scalable Video Coding (SVC) for the Home Entertainment Division (HED) of STMicroelectronics.

4)   I have led the AST effort on OpenMax 1.0 standard development with Khronos and the development of the OpenMax 1.0 reference source code (Bellaggio). The next evolution i.e. OpenMax 2.0 is also being developed in AST Noida under my management.

Projects done prior to AST assignment (prior to 2001)

5)   Implementation of a V.90(56Kbps) and V.34(33.6Kbps) compliant wireline modem on a Pentium P2 platform under WinNT & Win2000. The challenge of the project was in matching the stringent real-time requirements of modem operation with the non-real time performance of the Windows Operating System. I was the leader of this project and the project team had 7 members.

6)   Implementation of a HDTV receiver for DVB-terestrial television using the COFDM standard. I was involved in the design and implementation of the timing and frequency synchronization algorithm, as per CCETT recommendations, on a proprietary DSP

core.

7) Design and implementation of a DVD drive front end. I designed the specifications for the real time kernel, command parser and the servo control module. These three components formed the embedded system for the reception and interpretation of commands from the DVD drive back end and the servo actions corresponding to the commands. The kernel and the command parser were implemented on a ST10-166 microcontroller. Subsequently, I did the design and implementation of the servo compensator for the optical tracking and focus mechanism for the aforementioned DVD drive on a proprietary DSP core.

8) I worked for the Memory Products Group of STMicroelectronics-India (SGS-Thomsom Microelectronics) from Feb.1996-Jan 1997. I have had exposure to the design of asynchronous SRAMs and have been actively involved in the design of 2 fast SRAM parts.

9) I also contributed to a project dealing with Capacitive Discharge Ignition Control for combustion efficient, clean exhaust 2 wheeler vehicles. My role was to design the ignition control algorithm, which had very stringent accuracy requirements in terms of angle and duration of firing control. The patent application for this algorithm has been filed.

10) I was the leader of a project for the design of an intelligent battery charger for mobile phones. This was meant to demonstrate the capabilities of a new battery charging device from STMicroelectronics. I personally designed an inflection point detection algorithm for this project.

11) I have been the technical leader for the Smart Card activity of ST-Noida in India and have contributed actively in the Smart Card Federation of India, which is a standardization body for smart card applications, under the umbrella of the Ministry of Information Technology, Government of India

Academic Activities:
1) I have been holding an adjunct faculty position in Dept. of Electrical Engineering, IIT, Delhi, since 2004-05, where I teach two M.Tech/Pre-PhD. level courses on VLSI design. The course titles are
    i. Issues and Challenges in Deep Sub Micron VLSI Design.
    ii. Design and Testing of Semiconductor Memories
2) I have been instrumental in setting up the ST-IITD research collaboration between STMicroelectronics and Indian Institute of Technology, Delhi, India,

which is the first of its kind in the region.

3) I have been actively involved in the academic activities of Deptt of Electronics, Delhi College of Engineering (now DTU), through delivering tutorials, examination of M.E. students, curriculum formation meetings and advice to faculty.

KNOWLEDGE LEVEL:

(a) During M.Tech., I have done courses in digital design and signal processing, having obtained good grades and have done practical work in the areas. I have done well in courses on MOS L.S.I. design and V.L.S.I. CAD and have done useful software development in both.

(b) I have held teaching assignments at the Indian Institute of Technology, Delhi, India, where I teach advanced (Master's level) courses on VLSI design.

(c) I am very well conversant with SPICE and have been able to carry out modifications in the package obtain better numerical performance.

(d) As I have done Ph.D. research work in analog VLSI design (please refer to the annexure), I consider myself especially strong in the field.

(e) I have had good exposure to the design and testing of SRAMs.

(f) Strong background in DSP, especially in adaptive signal processing algorithms and analysis of speech and image signals.

(g) I have good practical and theoretical experience in designing systems with microcontrollers, from the h/w as well as s/w points of view.

(h) I have a strong software background, especially in C and C++ and in the theory and practice of operating systems.

(i) I have a good background in Mathematics, especially Linear Algebra, Numerical Analysis and Functional Analysis.

(j) I have a good background in parallel computing, specifically for the area of embedded applications, holding patents in the area.

PROJECTS UNDERTAKEN:

At STMicroelectronics (Since 1996 till date, descending chronologically)

(1) Development of Context Adaptive Binary Arithmetic Coding hardware IP using High Level Synthesis for ST-Ericsson Fairbanks platform

(2) Development, porting and profiling of Google Android software stack on ST-Ericsson MontBlanc platform over Linux-SMP OS

(3) Research and development of Power Management algorithms for ST-Ericsson MontBlanc platform over Linux SMP OS Power Management infrastructure

(4) Participation on OpenMax 1.0 standard development and development of OpenMax reference software (Bellaggio)

(5) Research and development of decoder architecture for the Scalable Video Coding (SVC) standard for Home Entertainment Division (HED) of STMircoelectronics.

(6) Development of parallel algorithms for MPEG audio/video decoding on Linux using MPI and shared memory programming (SMP) paradigms.

(7) Design and implementation of VDSL modem (50Mbps) on a VLIW architecture

(8) Design and implementation MPEG2AAC audio decoder on a VLIW architecture

(9) Design and implementation Dolby AC3 audio codec on a VLIW achitecture

(10) Design and implementation of a Softmodem with modulation protocols upto V.90.

(11) Design and implementation of the timing and frequency synchronisation algorithms for a HDTV receiver based on the COFDM DVB-Terrestrial standard.

(12) Design and implementation of the servo compensators for the tracking and focus mechanisms of a DVD drive.

(13) Design and specification of a DVD drive front end mechanism as a real time embedded system.

(14) Design of an intelligent battery charger for mobile phones.

(15) Capacitive Discharge Ignition Control Algorithm.

(16) Design of a fast (12ns read-write access) 128Kx8 asynchronous SRAM.

The following projects were done as a part of my Masters and Doctoral studies (1988-1995):

(17) Development of design strategies for analog neural networks. (Doctoral Thesis)

(18) Development of a partitioning method for reducing the connectivity of highly connected neural networks for easy I.C. realization. (Doctoral Thesis)

(19) Phoneme recognition from continuous speech using neural network (Masters Thesis).

(20) Speech compression to low bit-rates under the celp32 standard. (Project Guidance)

(21) Development of an adaptive strategy for lowering bit rates in MPEG compression. (Project Guidance)

(22) Speech compression to low bit-rates under the LDCELP and CELP32 standard (project Guidance).

(23) Modification of SPICE(3b) for improving solution quality in the case of ill-conditioned circuit matrices i.e. for MOS devices in weak inversion. (Doctoral Study)

(24) Development of an RTL using YACC and LEX under UNIX on HP-9000s300 machine. (Lab Project)

(25) Development of a package for module placement using simulated annealing method (Lab Project).

Patents

1. " METHOD AND APPARATUS FOR LINEARLY SCALABLE FFT/IFFT"---filed
2. "METHOD AND APPARATUS FOR LINEARLY SCALABLE FIR FILTER WITHOUT DELAY LINE"---Granted
3. "METHOD FOR MULTIPROCESSOR FFT/IFFT WITH MINIMUM INTER-PROCESSOR DATA COMMUNICATION OVERHEAD"--- Filed
4. "MEANS FOR ENHANCING PERFORMANCE OF FFT/IFFT UTILISING MEMORY LOAD/STORE UNITS WITH AUGMENTED DATA BUS INSTEAD OF MULTIPLE MEMORY LOAD/STORE UNITS "---filed
5. " PARALLEL IMPLEMENTATION OF MPEG-2 VIDEO DECODER IN DISTRIBUTED MEMORY SYSTEM" – filed
6. LOW COMPLEXITY & LOW COST SOLUTION TO CLOCK RECOVERY PROBLEM IN THE PRESENCE OF LARGE CLOCK JITTER IN DVB-TERRESTRIAL – Granted
7. A SYSTEM AND METHOD FOR A FAST CLOCK RECOVERY IN DIGITAL VIDEO COMMUNICATION – Filed
8. CHANNEL SELECTION – filed
9. A METHOD FOR ESTIMATING NUMBER OF BITS TO ENCODE A MACROBLOCK AND A BIT ESTIMATOR – Filed
10. AN INNOVATIVE APPROACH FOR ADAPTATION OF GENERIC RATE CONTROL ALGORITHMS FOR TARGET VIDEO STANDARDS - Filed

Publications

1) Parallel Implementation of MPEG-2 Video Decoder, *Sarkar,A., Maiti, S., Saha, K. – IS&T / SPIE's Symposium on Electronic Imaging: Science & Technology, San Jose, Ca, USA, Jan 2006.*
2) Current and future trends in embedded VLIW microprocessors applied to multimedia and signal processing, *Desoli G., Strudel T., Cousin J-P., Saha K., Invited Paper, 14th European Signal Processing Conference, Florence, Sept 2006.*
3) Parallel Implementation of Dolby-AC3 Audio Decoder on Multi-Processor SMP platform, *Soni Y.K., Gupta A., Saha K., ST Journal of Systems Research, vol1,2007*
4) Scalable Integer based Frequency Error Estimation Technique for Clock Recovery in Packet Switched Networks, *Mathur M. and K., Internet and Multimedia Systems and Applications (IMSA 2007) , Aug 2007, Honolulu, Hawaii, USA*

5) An innovative approach for adaptation of generic rate control algorithms for target video standards, *Agarwal M., Johar S., Saha K., Piccinelli E., IEEE International Conference on Signal Processing and Communications (ICSPC 2007), Nov 2007, Dubai, UAE*

6) Pre-coded video transcoding into H.264 with arbitrary resolution change, *Tripathi S, Saha K., Piccinelli E., IEEE International Conference on Signal Processing and Communications (ICSPC 2007), Nov 2007, Dubai, UAE*

7) Real-Time SVC Decoder in Embedded System, Maiti S.N., Gupta A, Saha K., Piccinelli, E.M., *International Conference on Signal Processing and Multimedia Applications, SIGMAP 2009, July 2009, Milan, Italy*

8) AVC to SVC Decoding: Overhead Analysis in Real Time Embedded System, *Maiti S.N., Gupta A, Saha K., Piccinelli, E.M., International Conference on Signal Processing and Multimedia Applications, SIGMAP 2010, July 2010, Athens, Greece, (Paper communicated)*

PhD Theses supervised.

1) Studies in architectures and applications of low-power ad-hoc wireless networks - Kumar M., PhD Program in Electronics and Communination, BITS Pilani.
(Thesis in progress)
2) Architecture and Design of >10Gbps I/Os – Monga S., PhD Program in Electrical Engineerind, IIT-Delhi.
(Thesis in progress)

M.Tech / MS Theses supervised:

1) H.263 Encoder for Videoconferencing: Raizada R., MS in Software Systems, BITS Pilani, 2003

2) H.263 Decoder for Videoconferencing: Narula S., MS in Software Systems, BITS Pilani, 2003

3) Digital Set Top Box for Indian Market Requirements: Sharma M.K., MS in Software Systems, BITS Pilani, 2004

SOFTWARE KNOWLEDGE:
(a) Well versed in C++, Pascal, Fortran, Lisp and the theory and practice of operating systems.
(b) Proficient in DSP algorithms and programming. Experience with

various brands and versions of DSPs
(c) Methodologies for architecture space exploration for advanced CPU design.
(d) Linux and MS Windows device driver development
(e) Good background in computer simulation and simulator development.

Foreign langauges spoken: French (Fluent), Italian (Fluent),German (Intermediate)

APPENDIX 2

SRIJIB NARAYAN MAITI

Contact Address
22B, Express View Apartment, MIG,
Sector-93, Noida-201304, India.
E-mail: srijibmaiti@gmail.com. **Telephone:** 00-91-9810403772(Mobile).

Permanent address
R-5/1, Duk Bungalow Road, Saratpalli,
Midnapore (W), West Bengal-721101, India

Objective: To excel in embedded technologies through R&D activities in multimedia and parallel processing.

Total Industrial Experience : ~ 13 years. **D.O.B.:** 24.10.1975 **Sex:** M

Present Status: Working with **STMicroelectronics Pvt. Ltd., Inida, in Advanced System Technology group.**

Experience Summary	Major Achievements
-- R&D in **parallel processing, multimedia codecs (MPEG2, H.264, SVC, MVC) architecture design** and implementation. -- Experienced in working with ST200 (Family of **VLIW processor**), optimization (up to assembly level), Architecture Evaluation, Feasibility and porting of Multimedia Applications and **DSP** Algorithms. -- Design and development experience in DLC (Digital Loop Carrier) system (PSTN). -- Implementation knowledge about **CMM, ISO.** -- Work as **patent committee member** to evaluate patent proposals -- **ISO/ TS 16949** auditor.	-- **6 patents on FFT, FIR, MPEG2, SVC** video decoding. -- Paper publications in International conferences.

Educational Qualifications

--**Masters** (M.E. in Electronics and Communications Engineering, DCE, 2004, 80%).
--**Graduation** (B.Tech in Electronics and Communications Engineering,
    R.E.C.Kurukshetra, 1997, 74.5%).

Employment History

Sr. no.	Establishment	Duration	Designation
1.	STMicroelectronics Pvt. Ltd.	Sept'2000 to till date	Technical Specialist
2.	HFCL(R&D)	Feb'99 to August 2000	Deputy manager, Assistant manager.
3.	Indian Telephone Industries(ITI) Ltd.	Oct'97 to Jan'99	Asst. Executive Engineer.

Project Experience

1. Multi View Coding (MVC) architecture design/product realization

- **Client:** STMicroelectronics internal
- **Description:** ST is in development of SoC which is capable of stereo MVC decoding for HD resolution. ST is one of the leading hardware IP (intellectual property) providers for H.264 decoding/encoding. This project is part of its future product development.
- **Responsibility:** Understand MPEG standard for MVC and find out the additional functionalities as compared to H.264 decoding. Analyze whether the existing architecture of H.264 decoding is capable of SVC decoding. Then, design architecture and implement MVC decoder in SoC.

2. Scalable Video Coding(SVC) architecture design/product realization

- **Client:** STMicroelectronics internal
- **Description:** ST is in development of SoC which is capable of SVC decoding for HD resolution. This project is also a part of its future product development.
- **Responsibility:** Understand and profile the SVC decoder starting from standard source code. Find out the additional functionalities as compared to H.264 decoding. Analyze whether the existing architecture of H.264 decoding is capable of SVC decoding. Then, design architecture and implement SVC decoder in SoC.
- **Achievements:**

    --**Patent filed** on SVC video decoder, titled **"A Method and System for Parallel SVC decoder"**.

    --**Paper** accepted in **SIGMAP2009**.

3. Parallelisation of H.264 Video decoder

- **Client** STMicroelectronics internal
- **Description:** Job is to find out a method to parallelize H.264 and implement it for multiprocessor platforms, especially on shared memory systems.
- **Responsibilities:** To work as a team leader as well as team member in design and development.

4. Parallelisation of MPEG-2 Video decoder

- **Client** STMicroelectronics internal
- **Description:** Task was to find out a method to parallelize the advanced video codecs and implement in multiprocessor platforms both in distributed memory and shared memory systems. Subsequently, integration of parallel (Pthread)

MPEG2 video decoder to a real time DVD player (MPlayer-0.92) by porting of this DVD player s/w on a SMP platform.

- **Responsibilities**: To work as a tem leader as well as team member in design development and simulation/testing.
- **Achievements**:

  --**Patent filed** on MPEG2 video decoder, titled **"A MACRO-BLOCK LEVEL PARALLEL IMPLEMENTATION OF VIDEO DECODER"**.

  --**Paper** accepted in **IS&T/SPIE's Electronic Imaging 2006**.

5. **Parallel algorithm development for DSP-FE functions of VDSL modem**

   - **Client**:   STMicroelectronics internal
   - **Description**: It is visualized that the ST200 family of Very large Instruction Word   (VLIW) processors, jointly developed by STMicroelectronics and HP Corporation, may be used as a replacement for conventional DSP processors for application in the area of Very High-Speed Digital Subscriber Line (VDSL) modems. STMicroelectronics had a chipset for the VDSL application, based on the Zipper-DMT standard. This chipset consists of the Analog front-end, DSP front-end (DSP-FE) and ATM processor. It was important that a programmable processor based alternative solution be explored, to deal with programmability and upgradeability issues. The motivation behind the work was to explore the possibility of implementing all or some of the DSP-FE functions on the Lx VLIW processor.
   - **Responsibilities**: To develop algorithm and simulate all the above computational processes for multi-processor systems like Lx family of VLIW processor.
   - **Major Achievements**

     --**Patent filed on FIR (Finite Impulse Response) filter, titled "Method and Apparatus for Linearly Scalable FIR Filter without Delay Line".**

     --**Patent filed on FFT/IFFT, titled "Method and Apparatus for Linear Scalable FFT/IFFT".**

6. **Development of parallel algorithm for FFT/IFFT in Multi-processor systems with distributed memory architecture**

   - **Client**:   STMicroelectronics internal
   - **Description**: FFT/IFFT algorithm is developed in such a way that the data communication overhead between processors are minimum and the overall execution time for a given order of FFT/IFFT is reduced in the order of number of processors. So a cluster of 4 PCs (with Linux as operating system) is set up to simulate the algorithm with LAM-MPI.

- **Major Achievements**

   --Patent filed, titled "Method for Multi-processor FFT/IFFT with Minimum Inter-processor Data Communication Overhead".

   --Patent filed on FFT/IFFT, titled "Means for Enhancing Performance of FFT/IFFT Utilizing Memory Load/Store Units with Augmented Data bus Instead of Multiple Memory Load/Store Units ".

7. **Architecture Design and Development of subscriber line interface cards for Optical Access Networks (DLC).**

- **Client:** HFCL (Himachal Futuristic Communications Ltd.) internal.
- **Description:** Various types of subscriber lines viz. POTS, E&M, 2-Wire loop, ISDN-BRI, ISDN-PRI, 64Kbps co-directional data, Trunk line, E1 leased, HDSL, ATM, ADSL are terminated in one line card in Remote Terminal(RT). There are STM-1 ring running among the RTs. The bearer data is transported to the Central office (CO). The signaling date is processed and sent to the CO through the V5.2 protocol.
- **Line Cards developed:** ISDN-BRI, ISDN-PRI, E1 leased lines.
- **Responsibilities:** Detailed analysis of system followed by preparation of functional requirement specifications for ISDN-BRI card. Design of circuit schematics for the above cards, followed by preparation of software requirement specifications and design.